

# Esame di Calcolatori Elettronici T

## 9 Gennaio 2017 (Ing. Informatica)

### Esercizio 1

Progettare un sistema basato sul microprocessore DLX dotato di **128 MB di EPROM** mappata negli indirizzi bassi e **32 MB di RAM** mappata negli indirizzi alti. Nel sistema sono presenti **4 periferiche a 8 bit**, già progettate, denominate **INPUT\_PORT**, in grado di ricevere dati dall'esterno utilizzando il protocollo di *handshake*, e tre porte in output denominate rispettivamente **OUTPUT\_PORT**, **OUTPUT\_PORT\_PLUS\_1** e **OUTPUT\_PORT\_PLUS\_2** ciascuna in grado di inviare dati verso l'esterno utilizzando il protocollo di *handshake*.

**All'avvio debbono essere abilitati i trasferimenti da INPUT\_PORT e disabilitati i trasferimenti verso l'esterno.** Dopo il trasferimento in input deve seguire, **appena possibile**, un trasferimento in output verso **OUTPUT\_PORT** e, **se possibile e contemporaneamente a tale trasferimento**, anche verso le periferiche **OUTPUT\_PORT\_PLUS\_1** e **OUTPUT\_PORT\_PLUS\_2** che risultano pronte. Il dato da inviare in output è lo stesso letto in precedenza dalla porta in input e memorizzato nel **registro R20**. In seguito al trasferimento in output appena descritto deve essere riabilitato quello da **INPUT\_PORT**, disabilitando quello verso le porte in output e così via. Il processo di alternanza continua tra trasferimenti in input e output appena definito deve essere realizzato, **in modo automatico**, da **un'opportuna rete logica**. Tutte le periferiche presenti nel sistema saranno utilizzate unicamente per le finalità indicate nel testo.

- **Descrivere sinteticamente il progetto** che s'intende realizzare **evidenziando eventuali criticità**
- Progettare il sistema e le **reti logiche necessarie** per soddisfare le specifiche del problema **evitando clock-gating** e **utilizzando i segnali asincroni solo per inizializzare i dispositivi**
- Indicare le espressioni di decodifica e il range di indirizzi di tutte le periferiche, le memorie e i segnali
- Scrivere il codice dell'*interrupt handler*, **commentando in modo chiaro ogni istruzione**, che consente di gestire il problema in accordo alle specifiche assumendo che i registri da R20 a R30 possano essere utilizzati senza la necessità di dover preservare il loro contenuto (R20 dovrà essere utilizzato per gli scopi indicati in precedenza)

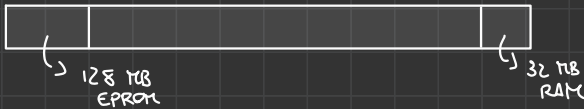
### Esercizio 2

Descrivere, con un esempio, la differenza tra il mapping di periferiche a 8 bit (con 4 registri interni) a indirizzi contigui e non contigui nel caso di un processore con bus dati a 32 bit.

### Esercizio 3

Descrivere il principio di funzionamento del *branch target buffer*. Dove e come agisce? Quali sono i problemi nel caso di loop annidati e quali strategie possono essere adottate in tal caso per aumentare le prestazioni?

## MAPPING



EPROM - 128 :  $0 \times 00000000 \rightarrow 0 \times 07FFFFFF$  ( $4 \times 32$ )

RAM - 32 :  $0 \times FE000000 \rightarrow 0 \times FFFFFFFF$

## CHIP SELECT

CS - EPROM - 0 =  $\overline{BA31} \overline{BA30} BE0$

CS - EPROM - 1 =  $\overline{BA31} \overline{BA30} BE1$

CS - EPROM - 2 =  $\overline{BA31} \overline{BA30} BE2$

CS - EPROM - 3 =  $\overline{BA31} \overline{BA30} BE3$

CS - RAM - 0 =  $\overline{BA31} BE0$

CS - RAM - 1 =  $\overline{BA31} BE1$

CS - RAM - 2 =  $\overline{BA31} BE2$

CS - RAM - 3 =  $\overline{BA31} BE3$

CS - INPUT - PORT =  $\overline{BA31} \overline{BA30} BE0 \cdot \overline{IBF} \cdot \overline{EN\_INPUT}$

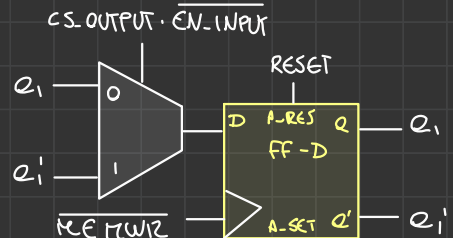
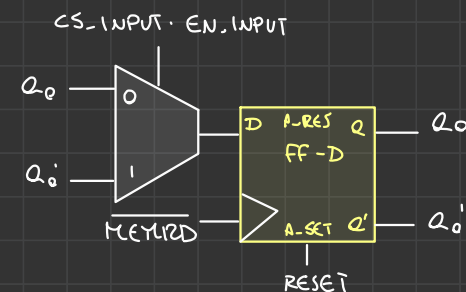
CS - OUTPUT - PORT =  $\overline{BA31} \overline{BA30} BE1 \cdot \overline{OBF} \cdot \overline{EN\_INPUT}$

CS - READ - STATUS =  $\overline{BA31} \overline{BA30} BE2 \cdot \overline{MEMRD}$

CS - OUTPUT - PLUS\_1 = CS - OUTPUT - PORT  $\cdot$  READY - PLUS\_1

CS - OUTPUT - PLUS\_2 = CS - OUTPUT - PORT  $\cdot$  READY - PLUS\_2

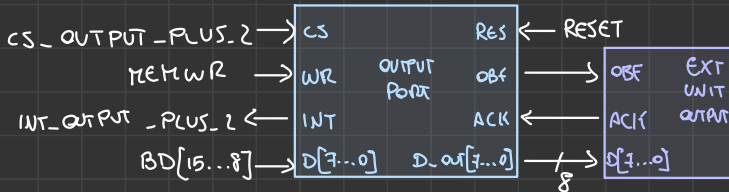
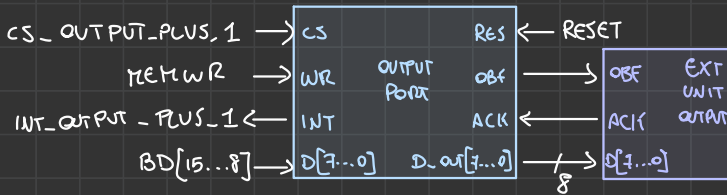
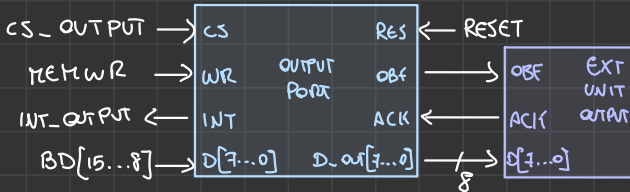
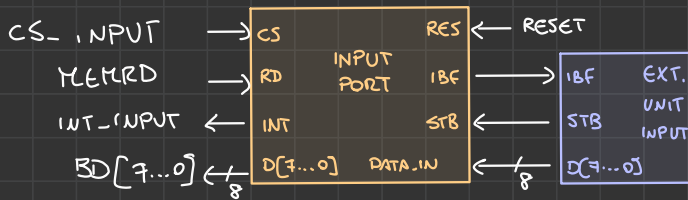
Reti per gestire l'alternanza tra INPUT e OUTPUT



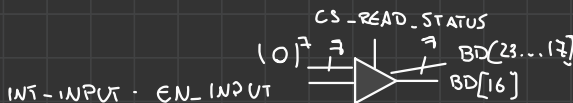
$Q_0$	$Q_1$	
0	0	OUTPUT
0	1	INPUT
1	0	INPUT
1	1	OUTPUT

$$\Rightarrow EN\_INPUT = Q_0 \text{ XOR } Q_1$$

# PORTC I/O



$$INT(TO\ DLX) = INT\_INPUT \cdot EN\_INPUT + INT\_OUTPUT \cdot \overline{EN\_INPUT}$$



## CODICE

```
0h    LHI    R21, 0x4000
4h    LBU    R22, 0x0002(R21)
8h    BEQZ   R22, INPUT
ch    SB     R20, 0x0001(R21)
10h    RFE;
14h    INPUT LBU    R20, 0x0000(R21)
18h    RFE;
```